User manual

January 8, 2011

by

created by w2u from User_manual.xml
(http://wiki.sparemint.org/index.php/Main_Page)

Contents

1 What is FreeMiNT? 2

2 How does FreeMiNT work? 2

3 Installation 2

4 The unified filesystem (u:\) 2

5 Memory protection 2

6 The MiNT system folder 3

7 Configuration 3
7.1 The boot menu . . . 3
7.2 The configuration files . . . 4
7.3 Keyboard layout . . . 4

8 Filesystems and filesystem drivers 4
8.1 VFAT and FAT32 . . . 5
8.2 ext2 . . . 5
8.3 minix . . . 6
8.4 nfs . . . 6

9 Device drivers 6

10 System extensions 7
10.1 MiNTnet . . . 7

11 XaAES 7

12 Special considerations 7

1 What is FreeMiNT?

FreeMiNT is a free, open-source operating system for 16/32bit Atari computers and clones. It is compatible with TOS, but adds a lot of new functionality not available in any other TOS-compatible operating system.

- Allows any number of GEM (with XaAES running) and TOS programs to run concurrently.
- Has memory protection to improve security and stability.
- Has built in network support and drivers for all existing networking hardware.
- Multi-user support.
- Has a wide array of system extensions to ease porting of unix programs.

2 How does FreeMiNT work?

FreeMiNT is a TSR that replace the GEMDOS -part of TOS. It also replace most of BIOS and XBIOS. The ROM AES can be replaced by the FreeMiNT kernel module XaAES or by some other AES like N.AES .

FreeMiNT implements all features from TOS. It also adds a lot of it's own, like networking, multiuser support, process control and most importantly the ability to run several programs concurrently.

3 Installation

See the installation guide for details on how to install FreeMiNT.

4 The unified filesystem (u:\)

5 Memory protection

A feature of FreeMiNT that is new to TOS-like systems is the concept of protecting the program's memory space from unauthorized "intruders". Such an intruder is usually another program, that went bad due to its bugs.

NOTICE: The memory protection mechanism works only on machines equipped with processor 68030 or newer.

The hardware memory protection is implemented at the CPU level on a per process basis, so that any memory a process hasn't allocated is considered invalid for that process. Virtually, such memory does not exist, and starts to do just after the process requests it from the system. The memory protection serves for preventing an errant process from destroying other programs running in parallel, or even the system itself. At least, that's the theory.

In practice, apart from running Un*x software (which usually is perfectly compatible with memory protection) FreeMiNT has to be compatible with TOS software. Unfortunately some TOS programs - especially older applications and MIDI software - tend to enter supervisor mode and do things beyond the system control.

Under TOS it is perfectly okay. However, under FreeMiNT there's a risk for a conflict with the resource protection mechanisms. This can cause the program or even the entire system to crash.

Memory protection is on by default, but can be switched off in the boot menu.

Fine-tuning and troubleshooting memory protection.

6 The MiNT system folder

FreeMiNT expects to find a folder named **mint** in the root of your boot-drive. This folder contains the various config-files, modules and tools required by MiNT.

Optionally, this can be located in subfolders named after the kernel version number. For example FreeMiNT 1.17.0 will look for the subfolder '1-17-0/', and if this folder doesn't exist it will load the config file and modules located in the root of the system folder. This allows the user to keep separate system folders for different versions of the kernel.

So if your boot-drive is C: and you boot FreeMiNT 1.17.0, it will look for the following folders:

c:\mint\1-17-0

c:\mint

It will load it's config-files and modules from the first folder it finds.

Exception: If you're using a beta or alpha version of the kernel, it will look for a special "beta"-folder. This will have the suffix "-cur", e.g. '1-17-cur' for the current alpha or beta of FreeMiNT 1.17. This will allow you to have one system folder for the latest beta, and one for the release version.

Warning: since version 1.16.0 FreeMiNT ignores **multitos** folder in the root of your boot-drive (done for backward compatibility with MultiTOS) !

7 Configuration

7.1 The boot menu

Hold down the left shift key while booting FreeMiNT to enter the boot menu. There are nine options in this menu, all of them can be toggled by pressing the key next to the menu entry.

- Start up FreeMiNT: Would normally be set to **yes** (for obvious reasons). If you don't use a boot manager, you can use this feature when you don't want/need to boot FreeMiNT.
- Load external XFS: Normally set to **yes**. If set to **no**, filesystem drivers are not loaded. You will then not be able to access other filesystems than FAT and FAT32.
- Load external XDD. Normally set to **yes**. If set to **no**, no device drivers will be loaded.
- Execute AUTO PRGs: Normally set to **yes**. If set to **no**, FreeMiNT will not execute programs located after FreeMiNT itself in the autofolder.
- Memory protection: If **yes** then memory protection will be enabled (recommended).
- Init step by step: If set to **yes**, booting will be paused after each step until you press a key. Useful when debugging the boot sequence.
- Debug/trace level: Applies only to debug kernels. Sets debug level.
- Debug output dev.: Applies only to debug kernels. Select debug output device.
- Remember settings: If set to **yes**, the selections you've made in the boot menu will be saved to mint.ini.

The boot delay (the amount of time FreeMiNT waits for the user to enter the boot menu) can not be set in the boot menu. If you want to change this delay, you must edit mint.ini with a text editor.

7.2 The configuration files

There are three configuration files in FreeMiNT. All of them are located in the system folder. These files are only read when the system starts, so for any changes to take effect the system must be restarted.

mint.ini This file is automatically created by FreeMiNT, and contains the settings in the boot menu. It can be manually edited, but any comments will be deleted the next time FreeMiNT saves it's boot-settings.

mint.cnf This file contains a set of commands and variables that determines how FreeMiNT will behave when started. The installation tool will set up some usable defaults, but if you want to tune your system this file is where you start.

xaaes\xaaes.cnf Similar to **mint.cnf**, this file contains settings relevant to XaAES.

7.3 Keyboard layout

If your language is not directly supported by TOS, or if you use non-standard keyboard (e.g. a PC- or Mac-keyboard) you will probably need to modify the keyboard layout in FreeMiNT to match the physical layout of your keyboard. Traditionally this has been done by a small auto folder TSR that installs a new keymap. This will also work with FreeMiNT, but a safer way of remapping the keyboard is to let FreeMiNT do it by loading a keyboard layout file during boot. If the file **keyboard.tbl** is present in the system folder , it will be used as the default keyboard layout in FreeMiNT.

This has a few advantages over the traditional TSR-solution:

- Clean and safe.
- The ALT-key can be used as a modifier.
- On the Milan and under ARAnyM , the AltGr-key can also be used as a modifier.
- Deadkeys are supported, very handy if your language has more than a handful of accented characters.

Keyboard tables can be generated with the **mktbl** utility. If you prefer a GEM-tool, KeyEdit can create, edit and install FreeMiNT keyboard tables.

Since kernel 1-17-0, keyboard.tbl files for several different keyboard layouts (ready to use) can be found in the FreeMiNT build.

- Read more about the Keytable file format.

8 Filesystems and filesystem drivers

Filesystems are implemented as kernel modules. Filesystem kernel modules can be identified by the .xfs file extension. Usually you just have to place the filesystem kernel module into the system folder to enable support for an specific file system.

Currently supported filesystems:

- ext2 (ext2.xfs)
- minix (minix.xfs)
- fat/vfat/fat32 (integrated into the kernel)
- nfs (nfs.xfs)

When you partition a hard drive for use with GEM, all partitions are usually set up with the FAT filesystem. If you want to use other filesystems, you have to do two things:

1. Copy the required filesystem driver to the system folder.
2. Initialize at least one partition with the new filesystem.

Case sensitivity
File and Path name Lengths

8.1 VFAT and FAT32

Support for these FAT extensions is integrated in the kernel, no separate driver is necessary. VFAT is an extension to the FAT filesystem that allows long filenames to be used. This can be used on all existing FAT partitions, one needs to allow VFAT extension for each drive in mint.cnf though. It's not recommended to use VFAT-aware drives later in TOS as this can lead to filesystem corruption.

FAT32 is similar to FAT(16), but can handle much bigger files and partitions. You need to initialize at least one partition using mkfatfs or HD-Driver . The partition type must be set to 'F32' using a partitioning tool to use this filesystem. FAT32 partitions can not be accessed from TOS.

VFAT/FAT32 in FreeMiNT is fully compatible with Windows VFAT/FAT32, so if you use a harddisk driver that understands PC-partitioned disks you can directly attach a disk formatted under Windows as FAT32 to Atari and read / write files from / to it. MagiC offers similar option but it's much slower with accessing such drive.

Limits

maximum partition size FAT16 : 2 GB FAT32 : > 16 TB maximum file size : 2 GB block size : 512 byte - 32 kb physical blocksize must be smaller or equal to logical blocksize : yes

8.2 ext2

This filesystem is used by linux. It has several advantages over FAT, and if you want to install e.g. EasyMiNT it's mandatory. To use this filesystem perform the following steps:

1. Copy ext2.xfs to the system folder.
2. Use a partitioning tool to set the partition type for the desired partition to 'lnx' or 'raw'.
3. Create the filesystem with mke2fs.
4. Reboot.

Limits

maximum partition size : > 16 TB maximum file size : 2 GB block size : 1, 2, 4 kb physical blocksize must be smaller or equal to logical blocksize : yes

8.3 minix

Made obsolete by ext2. If you want to use this filesystem anyway, the procedure is similar to ext2.

1. Copy minix.xfs to the system folder.
2. Use a partitioning tool to set the partition type for the desired partition to 'mnx' or 'raw'.
3. Create the filesystem with minit.
4. Reboot.

Limits
maximum partition size : 2 GB maximum file size : 2 GB block size : 1 kb physical blocksize must be smaller or equal to logical blocksize : yes

8.4 nfs

A networking filesystem. Copy nfs.xfs to the system folder to enable it. This filesystem is used to mount NFS or Samba shares on a network.

9 Device drivers

Device drivers are kernel modules that provides the operating system and/or user applications with an interface to a specific piece of hardware. The drivers are located in the system folder. Currently there are two types of hardware drivers:

Hardware drivers These drivers are recognized by their *.xdd filename extension. One driver usually supports one specific piece of hardware, and installs a device for it in the **u:\dev** directory. This device can then be accessed by user programs. It is advisable to only enable the drivers for installed hardware. To disable a driver, rename the filename extension to something else than .xdd.

The following device drivers are a part of the FreeMiNT project:

- inet4.xdd - TCP/IP Networking for FreeMiNT
- audiodev.xdd - /dev/audio for all Ataris.
- dsp56k.xdd - Falcon DSP Device driver.
- flop_raw.xdd - Raw floppy access.
- lp.xdd - Parallel interface driver.
- mfp.xdd - MFP Serial Port driver, for systems having MFP serial ports (ST, Mega ST, Mega STE, Hades, FireBee).
- scc.xdd - SCC Serial Port driver, for systems having SCC serial ports (Mega STE, Falcon, TT).
- uart.xdd - 16550 UART Serial port driver (Milan).
- xconout2.xdd - implements u:/dev/xconout2. This must be installed if you use TosWin2 's console window.
- dummy.xdd - Just a test driver, no need to install.

- inet4.xdd - TCP/IP interface. This is an exception from the other drivers. It does not provide a hardware driver directly, but implements the TCP/IP networking protocol.

Network hardware drivers These drivers are recognized by their .xif filename extension. They are loaded by the inet4.xdd hardware driver, and provides driver routines for various networking hardware. It is advisable to only enable the drivers for installed hardware. To disable a driver, rename the filename extension to something else than .xif.

The following network hardware drivers are included in the FreeMiNT project:

- ethernat.xif - Network Driver for EtherNAT hardware.
- de600.xif - Driver for the DE600 ROMport ethernet adapter.
- lance.xif
- pamsdma.xif
- biodma.xif - Driver for the BioData ACSI ethernet adapter.
- riebl*.xif - Suite of drivers for the various Riebl ethernet adapters.
- rtl8012.xif - Driver for the Genius ROMport ethernet adapter.
- scsilink.xif - Driver for Daynaport SCSI ethernet adapter.
- nfeth.xif - Network driver for ARAnyM.
- slip.xif - Driver providing SLIP and PPP over serial links.
- plip.xif - PLIP driver for the parallel port. Allows two Ataris to be networked over the printer port. NB! This PLIP protocol is not compatible with the PLIP protocol as implemented under Linux.

Third party network drivers.

10 System extensions

10.1 MiNTnet

11 XaAES

XaAES is a FreeMiNT kernel module which replaces the AES.

12 Special considerations

How to run MiNT on "odd" hardware.